International Journal of Applied Engineering
Research and Development (IJAERD)
ISSN(P): 2250-1584; ISSN(E): 2278-9383
Vol. 4, Issue 2, Apr 2014, 57-64

TRANS STELLAR
• Journal Publications • Research Consultancy

HORIZONTAL AGGREGATION- GOLD IN COAL MINE

SHILPA KADAM[1], VIJAY BHOSALE[2] & UMESH KULKARNI[3]

[1,2]Department of Computer, MGM College of Engineering and Technology, Mumbai, Maharashtra, India

[3]Department of Computer, Vidyalankar Institute of Technology, Mumbai, Maharashtra, India

ABSTRACT

Datasets are widely used data mining, with columns in horizontal tabular layout to efficiently analyze the data. To extract trends and pattern from the historical data to prepare the future strategies, Data mining is the widely used domain. And this is a tedious task as it requires many complex queries, aggregation columns and joining tables. For statistical analysis, information gathered and represented in a summary form in data aggregation process. With one variable per column, surplus records plus horizontal layouts with data sets are the requirement of many algorithms. Data mining tools will create a SQL code representing horizontal aggregation as a template is the first.

Advantage of horizontal aggregation, which cuts shorts the human work in data preparation mode of data mining. End user written SQL code is inefficient then the code generated automatically is the second advantage plus this takes less time. Next advantage is that the data sets can be created in the DBMS.

KEYWORDS: Aggregation, SQL (Structured Query Language), SPJ, PIVOT, CASE, Data Set

INTRODUCTION

Vertical aggregation functions in SQL are the SUM, MIN, MAX, COUNT and AVG, these functions gives single row output on the column in the table. Aggregation will always support to give us the summary of data. Results of vertical aggregation can be fruitful for computation purpose but cannot for data mining operations [2]. Many data mining algorithms like PCA, regression, classification and clustering expects one dimension per column, data sets with a horizontal layout and several records as inputs [1]. Without DBMS, it is a difficult task to manage the large data sets. Using operator will help in performing horizontal aggregation same as select, project and join inside a query processor. There are three horizontal aggregation operators PIVOT, SPJ and CASE. Data analysis, data presenting, exchange

Rows and enable data transformation in data modeling is easily done with PIVOT operator in tabular data [1]. As horizontal aggregations are capable of producing data sets that can be used for real world data mining activities. With help of construct with standard SQL operation SPJ aggregation is developed. Built in pivoting facility provided by SQL is used in PIVOT while based on the SQL CASE construct the CASE, works on. SPJ provides condition output while the PIVOT does not provide output of any such fashion. SPJ is simple to implement but very slow in performance while Pivot is faster in performance but difficult to build. CASE is a combination of SPJ and PIVOT. It picks up the advantages of both the method to build its own. Combined with GROUP BY clause and the required number of columns to store the transposed table is decided by the PIVOT method [3]. CASE operator is evaluated by GROUP BY and CASE statements. Space occupied by the end user details can be reduced by this method [4].

SELECT columns, Aggregate Function

(CASE WHEN Boolean expression THEN result

ELSE result expression END)

FROM table GROUP BY column.

In SPJ method all tables are joined to create a table containing horizontal aggregation [4]. The difficulty and time consuming are the two factors behind developing data sets for data mining, which give a motivation to horizontal aggregation. For this reason we have extended the functionalities of CASE, SPJ and PIVOT operators in such a way that they produce data sets with horizontal layouts.

RELATED WORK

From scientific data to business transaction, from satellite pictures to military intelligence is far huge to handle. To take decision by retrieving information is simply not enough anymore now. Xiang Lian and Lei Chen [5] analyzed cost models for evaluating dimensionality reduction in high-dimensional Spaces. In this model to evaluate the query performance over reduced data sets, methods like GDR, LDR, ADR is used. Specifically a novel (A) LDR method uses partitioning based on Randomized Search (RANS). LDR approach, PRANS, which is based on the cost model and can achieve good query performance in terms of the pruning power. Exhaustive experiments has proved the correctness of this cost model when compared to the existing LDR method and resulting partitions with low cost query. C.Ordonez [6]

Invented a way to efficiently compute fundamental statistical models inside a DBMS by exploring UDFs (User Defined Functions). The quadratic sum of cross products of points and linear sum of points, these two summary matrices on the data set are reseeding. The straight forward translation of K- shown mathematically and essential for all models. Scoring the data sets can be achieved by introducing efficient SQL queries to compute summary matrices. Introduced UDFs that works in a single table scan is based on SQL framework. Scoring data sets is accomplished by a set of primitive scalar UDFs and aggregated UDFs to compute summary matrices for all models. C.ordonez [7] had put forwarded a technique combination of K-means clustering with relational DBMS using SQL.

Three SQL implementation makes this technique works. First step is a straightforward translation of K-means computations into SQL, and an optimized version based on improved data organization, efficient indexing, sufficient statistics, and rewritten queries, and an incremental version that uses the optimized version as a building block with fast convergence and automated means computation into SQL, works as a framework to build a second optimized version with superior performance. To introduce an incremental K-means implementation with faster convergence and automated reseeding this optimized version is taken as a building block. Conor Cunningalam [8] using two operator he came up with a technique of Optimization and Execution strategies in a RDBMS.

The two operators are PIVOT on tabular data which exchanges rows and columns which transforming data helps in data analysis, data presentation and data modeling. Implementing inside a query processor system is easy just like select, join and projector operator. Both during query execution and query optimization, such design provides better opportunities for better performance. Pivot is an extension of Group By with unique restrictions and optimization opportunities, and this makes it very easy to introduce incrementally on top of existing grouping implementations. H Wang C. Zaniolo [8] proposed a complete SQL Extension for Data Mining and Data Streams. A complete data intensive application in SQL can

be developed by writing table functions and new aggregates instead in procedural language. This technique is a powerful database language.

The ATLaS system consist of applications including various data mining functions, that have been coded in ATLaS‟ SQL, and execute with a modest (20–40%) performance overhead with respect to the same applications written in C/C++. Using the schemas of continuous queries and Query Repository Queries can be handled by this system.

AGGREGATION

To eliminate the limitations of SQL operator a new database language ATLaS was developed. ATLaS [11] can perform aggregations that are not possible with standard SQL. Only basic aggregation operations are performed by standard SQL. INITIALIZE, ITERATE and TERMINATE are the three functions into which the entire SQL statement is divided to perform operations in ATLaS. INITIALIZE section takes care of the declaration part, the ITERATIVE sections performs the major operation and the TERMINATE section takes care of the final statement to execute. It can also support online aggregation which is its major benefit. But its main drawback is that it consumes more space then the normal SQL plus results are not to be seen in horizontal tabular format.

Horizontal and Vertical Percentage

Using vertical and horizontal aggregation this method helps to calculate the percentages for the operations. Entire 100% results on the same row is seen in the A new class of functions which aggregates numeric expressions and the transposed results of data sets with horizontal layout, is brought by horizontal aggregation. A number of data mining tasks such as separately modeling and analyzing, segmentation of large mixed data sets into smaller uniform subsets and unsupervised classification and data summation needs operation. Efficient summary of data sets are needed to create data sets for data mining related works. Database by itself is a large amount of data and with the help of SQL information can be retrieved. Aggregation of huge amount of data is carried by SQL and aggregation helps to aggregate details of one table with another. Normal aggregation functions are sum(), min(), max(), count() and avg().

K	D1	D2	A
1	3	X	9
2	2	Y	6
3	1	Y	10
4	1	Y	0
5	2	X	1
6	1	X	NULL
7	3	X	8
8	2	X	7

Figure 1: Input Table

Vertical Aggregation

It is not at all different from standard SQL aggregation. It contains more number of rows and result is produced in vertical format. Some other methods can also produce results in vertical aggregate form.

D1	D2	A
1	X	Null
1	Y	10
2	X	8
2	Y	6
3	X	17

Figure 2: Traditional Vertical Aggregation

Horizontal Aggregation

The difference between vertical and horizontal aggregation is that horizontal produces result in horizontal tabular format. By help of any data mining tool data sets for all operations are produced and then on these data sets the aggregation operations are applied [10]. Small syntax extensions to normal SQL syntax is needed to produce the result in horizontal layout. The syntax for horizontal aggregation is given below.

SELECT columns, Aggregation Function

(Measure column BY Aggregating Parameters)

FROM GROUPING columns

In a horizontal aggregation there are four input parameters to generate SQL code:

- The input table F1,F2……,Fn
- The list of GROUP BY columns L1, . . ., Lj ,
- The column to aggregate (A),
- The list of transposing columns R1, . . .,Rk.

D1	D2 X	D2Y
1	NULL	10
2	8	6
3	17	NULL

Figure 3: Horizontal Aggregation

There is a common field K in Figure a and Figure b. In Figure b D2 consist X and Y distinct values used to form a transpose table. The sum () is used as the aggregate operation. The values within D1 are repeated, 1 appears 3 times, for row 3, 4 and, and for row 3 & 4 value of D2 is X & Y. So D2X and D2Y is newly generated columns in Figure c.

COMPARATIVE STUDY

Different methods performing horizontal aggregation are compared in the following

Grouping Combinations

To handle grouping of high dimensional data and the aggregation this operator was developed. All the drawbacks of normal grouping operator are overcome by this operator. ROLLUP, CUBE and GROUPING SET produce tabular results plus perform aggregation. If available data sets are huge this are not used. Long complex SQL queries can only help when huge input dataset is available. The ROLL UP operator can perform aggregation on smaller datasets and produce tabular results vertical format [4]. And this is not sufficient for mining methods. This drawback is overcome by the CUBE operator, which can perform aggregation even on large datasets. The main limitation of the CUBE operator is that it eliminates some of the details after performing aggregation. Hence GROUPING COMBINATION was developed to over the drawback of CUBE operator. Since, it operates on highly complex algorithms it performance is low while execution.

Atlas

To eliminate the limitations of SQL operator a new database language ATLaS was developed. ATLaS [11] can perform aggregations that are not possible with standard SQL. Only basic aggregation operations are performed by standard SQL. INITIALIZE, ITERATE and TERMINATE are the three functions into which the entire SQL statement is divided to perform operations in ATLaS. INITIALIZE section takes care of the declaration part, the ITERATIVE sections performs the major operation and the TERMINATE section takes care of the final statement to execute. It can also support online aggregation which is its major benefit. But its main drawback is that it consumes more space then the normal SQL plus results are not to be seen in horizontal tabular format.

Interpreted Storage Format

To handle the null values in both horizontal and vertical layouts the Interpreted Storage Format was developed. All the space data management complexities are taken care by this method. Large number of null values makes horizontal aggregation require more space. Nothing is for the null values in this method. A attribute identifier (attribute_id) and a length field is given as value in the row, when the row has value for an attribute. This value is stored with a particular head. This value is not easily accessible is its major disadvantage.

UNPIVOT Operator

Opposite to PIVOT is UNPIVOT operator which means transformation of columns into rows is done where which is exact opposite to the PIVOT operator. This in turn will increase the rows for the columns resulting in a big table. The mining algorithms which require horizontal table as input cannot use this method. This operator is commonly used for the statistical computation of some data mining

Approaches [12]. The general syntax is as follows:

SELECT columns

FROM table UNPIVOT

(Measure Column for Pivot Column IN (Pivot

Column Values))

Table 1: Comparative Study of Aggregation Methods

Methods	Aggregations	Feature
Grouping Combination Operator	Vertical and Horizontal	Complex Algorithms
Atlas	Vertical	Solves limitations of normal SQL
Vertical and Horizontal Percentage Aggregation	Vertical and Horizontal	Operates only on percentages
Interpreted Storage Format	Vertical and Horizontal	Difficult to retrieve data
Unpivot	Horizontal	Small syntax extensions in Select.

CONCLUSIONS

Using horizontal aggregation is main requirement for the retail industry. The strategy of business can be modeled keeping the results on horizontal aggregation in mind. This will in turn only help the business to grow. The horizontal aggregation results and proper business strategy will help to prosper. One more thing she be also taken care of is, which operator to use with which system.

REFERENCES

1. V. Pradeep Kumar1, Dr. R. V. Krishnaiah2 Dr. R. V. Krishnaiah2 Principal. Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis.

   IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 5 (Nov. - Dec. 2012), PP 36-41

2. Nisha.S*, B.Lakshmipathi Optimization of Horizontal Aggregation in SQL by Using K-Means Clustering Volume 2, Issue 5, May 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering Research Paper Available online at: www.ijarcsse.com

3. Mr. Prasanna M. Rathod ,Prof. Mrs. Karuna G. Bagde Workload Optimization by Horizontal Aggregation in SQL for Data Mining Analysis. ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology. (IJARCET) Volume 1, Issue 8, October 2012

4. Krupali R. Dhawale , Vani A. Hiremani Fundamental methods to evaluate horizontal aggregation in SQL.

   ISSN: 2278 – 7798 International Journal of Science, Engineering and Technology Research (IJSETR) Volume 2, Issue 10, October 2013

5. Xiang Lian, Student Member, IEEE, and Lei Chen, General Cost Models for Evaluating Dimensionality Reduction in High-Dimensional Spaces. *IEEE* Transactions on Knowledge and Data Engineering (TKDE), 22(1):139– 144, 2010.

6. C. Ordonez. Statistical model computation with UDFs. IEEE Transactions on Knowledge and Data Engineering (TKDE), 22, 2010.
7. C. Ordonez. Integrating K-means clustering with a relational DBMS using SQL.

   IEEE Transactions on Knowledge and Data Engineering (TKDE), 18(2):188–201, 2006.
8. C. Cunningham, G. Graefe, and C.A. Galindo-Legaria. PIVOT and UNPIVOT: Optimization and execution strategies in an RDBMS. In Proc. VLDB Conference, pages 998–1009, 2004.
9. H. Wang, C. Zaniolo, and C.R. Luo. ATLaS: A small but complete SQL extension for data mining and data streams. In Proc. VLDB Conference, pages 1113–1116, 2003.
10. M. Madhavi and S. Kavitha," Horizontal Aggregations in SQL to Prepare Data Sets for Data Mining Analysis", Madhavi etal. / IJEA, Vol. 1 Issue 6 ISSN: 2320-0804, PP 1-7, 2013.
11. K. Anusha, P. Radhakrishna and P. Sirisha," Horizontal Aggregation using SPJ Method and Equivalence of Methods", IJCST, Vol. 3, Issue 1, Spl. 5, pp 1-4, Jan. - March 2012 .
12. Mr. Ranjith Kumar K and Mrs. Krishna Veni," Prepare datasets for data mining analysis by using horizontal aggregation in SQL", Ranjith Kumar K et al, Int.J. Computer Technology & Applications, Vol 3(6), 1945-1949 IJCTA, pp. 1-5, Nov-Dec 2012.